(12) UK Patent Application (19) GB (11) 2 343 265 (13) A

(43) Date of A Publication 03.05.2000

(21) Application No 9823460.2

(22) Date of Filing 28.10.1998

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Carlos Francisco Fuente

(74) Agent and/or Address for Service
D P Litherland
IBM United Kingdom Limited, Intellectual Property Department, Mail Point 110, Hursley Park,
WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(51) INT CL[7]
G06F 11/10

(52) UK CL (Edition R)
G4A AME

(56) Documents Cited
EP 0482819 A2 EP 0462917 A2 US 5522031 A

(58) Field of Search
UK CL (Edition Q) G4A AME , G5R RB33 RGB
INT CL[6] G06F 11/10
Online: EPODOC

(54) Abstract Title
Data storage array rebuild

(57) A method is provided for preventing data loss during data reconstruction from a failed data storage device to a replacement data storage device in a redundant data storage array including a plurality N of data storage devices. In the array, data is arranged on the devices in multi-block stripes each of which comprises N-1 data blocks and a parity block, with one block from each stripe being located on each of the N devices. The normal reconstruction process includes reconstructing each data block of the failed storage device for each stripe in the array and storing the reconstructed data block on the replacement storage device. If during the rebuild process, a write I/O request to modify a data block is received and the request does not require access to the replacement disk, the write request is blocked, the data stripe which includes the data block to be modified is determined, the replacement data block for the determined data stripe is reconstructed for storage on the replacement disk; and the blocked write operation is restarted.

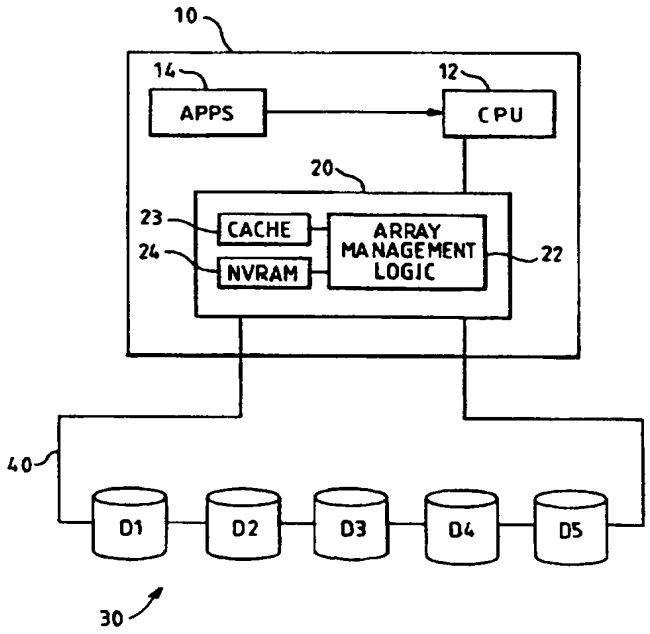

FIG. 1

GB 2 343 265 A

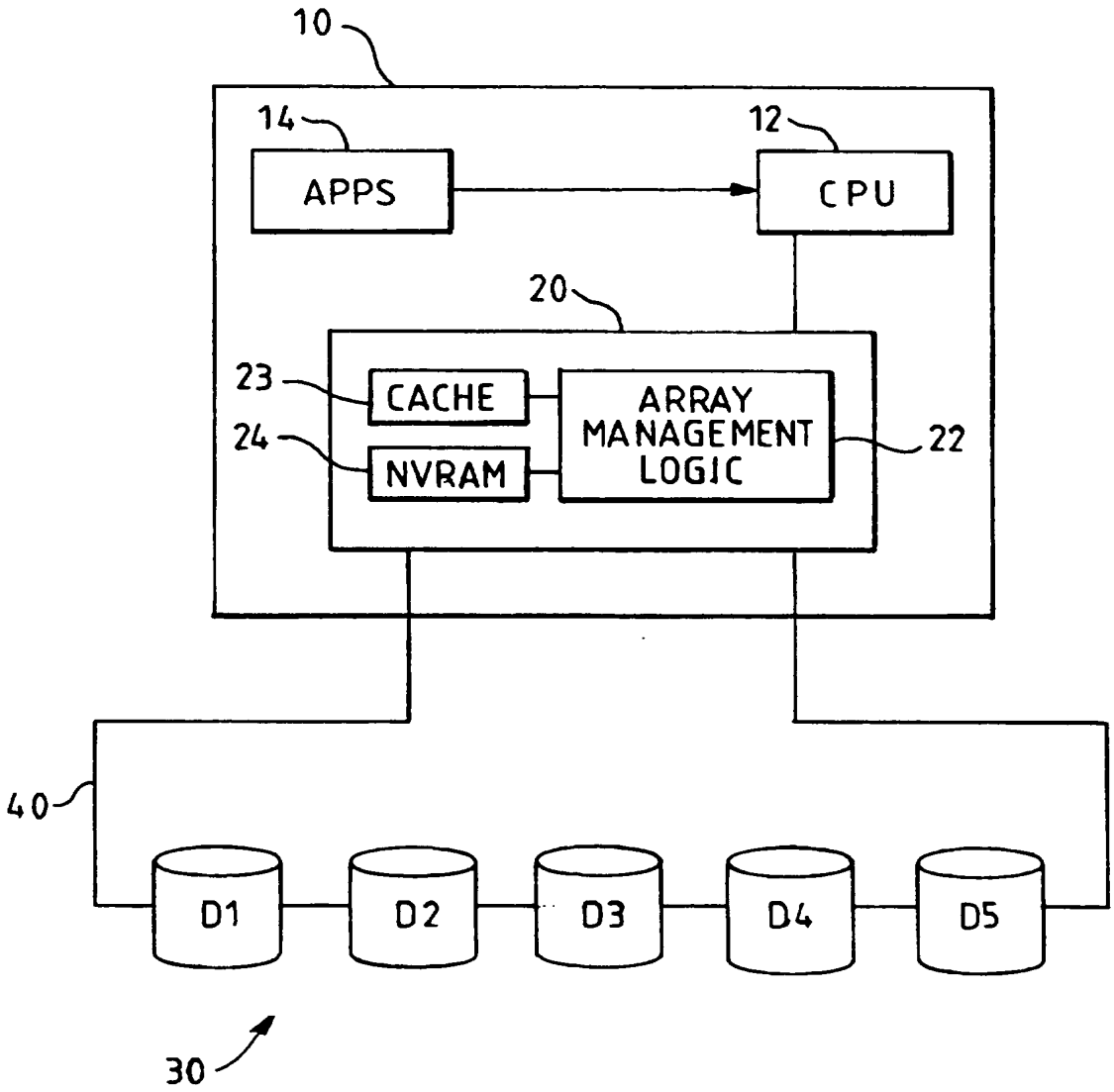

FIG. 1

	D1	D2	D3	D4	D5
A	1	0	1	1	(1)
B	1	1	0	(1)	1
C	1	0	(0)	0	1
D	0	(1)	0	0	1
E	(0)	1	1	1	1
F	0	0	1	1	(0)

FIG. 2A

	D1	D2	D3	D4	D5
A	1	0	1	—	(1)
B	1	1	0	—	1
C	1	0	(0)	—	1
D	0	(1)	0	—	1
E	(0)	1	1	—	1
F	0	0	1	—	(0)

FIG. 2B

	D1	D2	D3	D4*	D5
A	1	0	1	X	(1)
B	1	1	0	X	1
C	1	0	(0)	X	1
D	0	(1)	0	X	1
E	(0)	1	1	X	1
F	0	0	1	X	(0)

FIG. 2C

	D1	D2	D3	D4*	D5
A	1	0	1	1	(1)
B	1	1	0	(1)	1
C	1	0	(0)	0	1
D	0	(1)	0	X	1
E	(0)	1	1	X	1
F	0	0	1	X	(0)

FIG. 2D

	D1	D2	D3	D4*	D5
A	1	0	1	1	(1)
B	1	1	0	(1)	1
C	1	0	(0)	0	1
D	0	(1)	0	X	1
E	(0)	1	1	1	1
F	0	0	1	X	(0)

0

FIG. 2E

	D1	D2	D3	D4*	D5
A	1	0	1	1	(1)
B	1	1	0	(1)	1
C	1	0	(0)	0	1
D	0	(1)	0	X	1
E	(1)	0	1	1	1
F	0	0	1	X	(0)

FIG. 2F

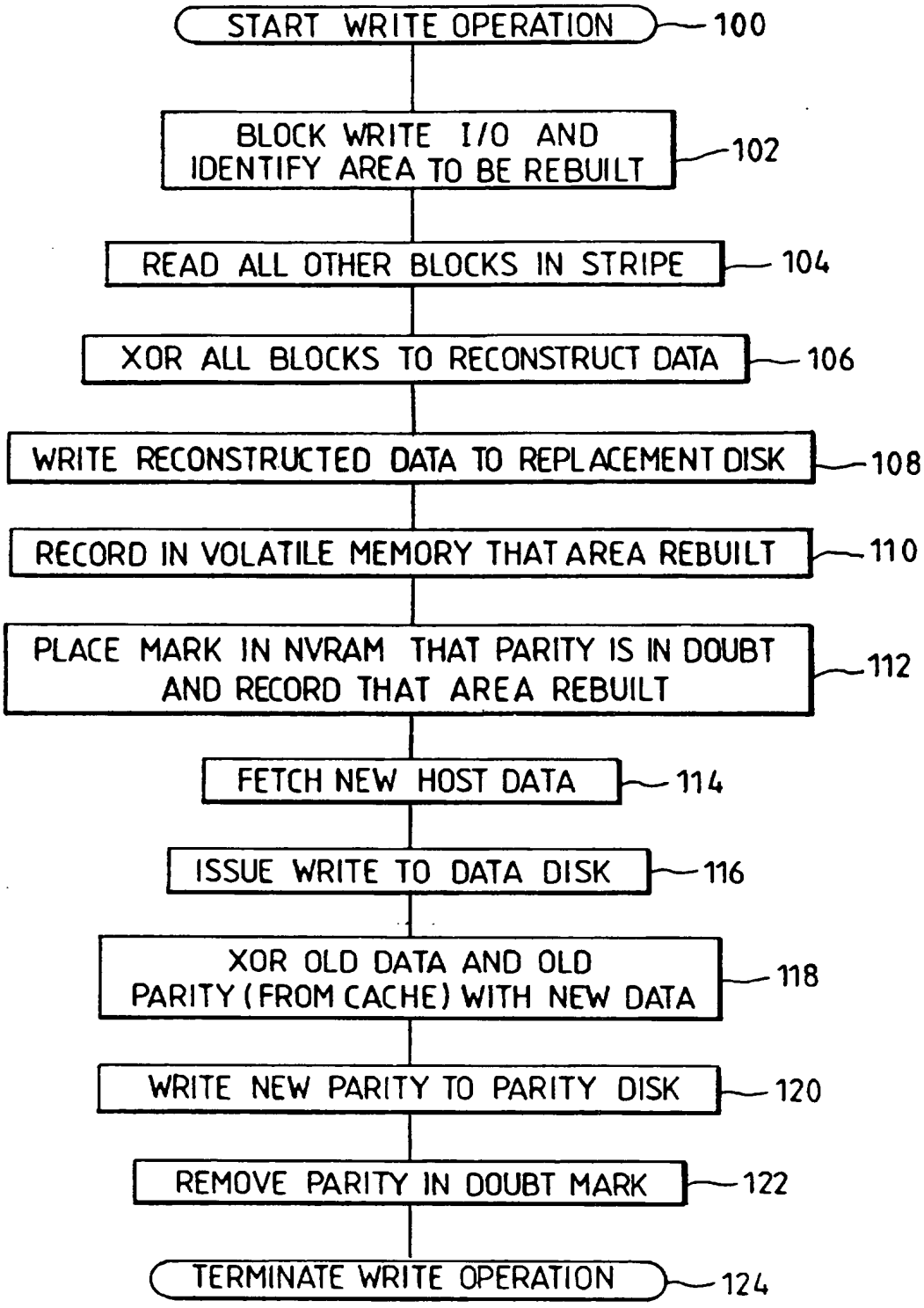

FIG. 3

2343265

DATA STORAGE ARRAY REBUILD

Technical Field of the Invention

The present invention relates generally to the field of data storage arrays and in particular to rebuild of a failed storage unit in a redundant data storage array.

Background of the Invention

Many of today's mid to high-end computer systems (for example network servers and workstations) include mass storage devices configured as a redundant array in order to provide fast access to data stored on the devices and also to provide for data backup in the event of a device failure. These arrays are commonly made up of a number of magnetic disk storage devices, which are held in an enclosure and connected to the host system by an array controller function which may take the form of either an array adapter located within the main processing unit of the computer system or alternatively a standalone array controller connected to the main processing unit. The interface between the main processing unit and the array often takes the form of one of the popular industry-standard protocols such as SCSI (Small Computer Systems Interface) or SSA (Serial Storage Architecture).

Storage arrays of this type are commonly arranged according to one or more of the five architectures (levels) set out by the RAID advisory board. Details of these levels can be found in various documentation including in the 'RAID book' (ISBN 1-57398-028-5) published by the RAID advisory board. Three of these architectures (RAID levels 3,4 and 5) are known as parity RAID because they all share a common data protection mechanism. Two of the parity RAID levels (4 and 5) are independent access parity schemes wherein a data stripe is made up of a number of data strips or blocks and a parity strip. Each data strip is stored on one member disk of the array. In RAID level 4, the parity strips are all stored on one member of the array. In RAID level 5, the parity strips are distributed across the member disks. In contrast with the parallel access schemes, an application I/O request in an independent access array may require access to only one member disk.

A limitation of the RAID levels 4 and 5 as compared to other levels is that writing a data block on any of the independently operating disk members also requires writing a new parity block onto the parity disk. The new parity is generated by, for example, XORing the old parity (read from the parity disk) with the old data (read from the appropriate disk) and the resulting sum is XOR'd with the new data. Both the new data and

new parity are then written to their respective disks. This process is often called a 'read-modify-write' (RMW) operation.

A key challenge in implementing independent access parity RAID lies in making the RMW operation sequence appear to applications as if it were a single write to disk. If a disk fails while a write operation is in progress e.g. after the new data is written to disk but before the updated parity is written, then parity and data will subsequently be inconsistent and in the event of a future disk failure, the data regenerated for that disk may be corrupted. The interval of time during which an array is susceptible to this form of data corruption is known as the write-hole. Independent access parity RAID arrays generally protect against data corruption due to write holes by keeping a log of write operations in progress in a small non-volatile memory (usually in NVRAM but alternatively on a disk).

As set out in detail in the above referenced 'RAID book', each RAID level provides for data protection in the event of disk member failure such that data continues to be available to applications. In RAID levels 4 and 5, the data strip from a failed disk can be reconstructed from the remaining data strips and parity strip held on the remaining disks. However whilst operating in this so-called degraded mode, the array does not provide the enhanced data reliability and availability of a fully functional parity RAID array. In order to restore full data protection the failed disk must be replaced by a functional one and the contents of the replacement disk be made consistent with the contents of the remaining array members. Making consistent the replacement disk's contents requires (i) reading corresponding strips (including parity) from each of the surviving original member disks; (ii) computing the XOR of these strips and (iii) writing the result to the replacement disk.

This process is called rebuilding or reconstruction and can take many hours for a replacement disk. In order to provide continued data availability, it is generally desirable to allow concurrent application I/O requests to the array while the array is carrying out the rebuild process. Two patents which describe on-line reconstruction of failed redundant array systems are US 5390187 and US 5522031.

There is the potential for data corruption if a write I/O occurs to an area of the array which has not yet been rebuilt on the replacement disk and power should fail. When power is restored the mark in non-volatile memory means that parity for the affected region of the array cannot be trusted. There are two possible outcomes:

(i) ignore the non-volatile mark - this means that the rebuild activity carries on regardless, treating the parity as good. This risks, in a

small percentage of cases, returning invalid data to the user, resulting in a miscompare, or verification error. This is generally unacceptable.

(ii) honour the non-volatile mark - this means that the rebuild must fail for the corresponding area of the replacement disk, and the array cannot be read for that area at all. This is comparable to a hard error from a normal disk and might result in the user needing to restore a file, filesystem or entire volume from backup.

Thus it can be seen that the combination of a rebuilding array and a power failure can result in a loss of data. It would be desirable to avoid such a problem.

Disclosure of the Invention

According to a first aspect of the invention therefore, there is provided a method for reconstructing data from a failed data storage device to a replacement data storage device in a redundant data storage array including a plurality N of data storage devices, data being arranged on the devices in multi-block stripes each of which comprise N-1 data blocks and a parity block, with one block from each stripe being located on each of the N devices, the method comprising reconstructing each data block of the failed storage device for each stripe in the array and storing the reconstructed data block on the replacement storage device; wherein in response to a write request to modify a data block that is part of a stripe which has not been reconstructed, the method comprises the further steps of: blocking the write request; determining the data stripe which includes the data block to be modified; reconstructing the replacement data block for the determined data stripe; storing the replacement data block on the replacement disk; and subsequently executing the write request.

It is preferred that method comprises the further step of making a record in non-volatile memory (e.g. NVRAM) that the replacement data block for the determined data stripe has been reconstructed and stored on the replacement disk. This non-volatile record must be maintained for the duration of the lifetime of the non-volatile parity in doubt mark. This record can be implemented in a number of ways. In a first, a separate record is made in NVRAM which is placed before, and removed after the parity in doubt mark. In an alternative, a flag or separate indicator may be used alongside the parity in doubt mark, so that the entry conveys both pieces of information explicitly. In a further alternative, the existing parity in doubt mark is used as an implicit indicator that the rebuild has been performed.

A record that the affected stripe has been rebuilt is also stored in volatile memory. This record is advantageously maintained until the whole of the replacement disk has been rebuilt in order to prevent multiple rebuilds of the same area on subsequent write I/O requests to that area.

According to a second aspect of the invention, there is provided an array controller for the reconstruction of data from a failed data storage device to a replacement data storage device in a redundant data storage array including a plurality N of data storage devices, data being arranged on the devices in multi-block stripes each of which comprises N-1 data blocks and a parity block, with one block from each stripe being located on each of the N devices, the controller including array management means for receiving, during the data reconstruction process, a write request to modify a data block which is part of a data stripe that has not been reconstructed, and responsive to such a request to block the write request, determine the data stripe which includes the data block to be modified; reconstruct the replacement data block for the determined data stripe for storage on the replacement disk; and recommence the write request.

According to a third aspect of the invention there is provided a data storage array comprising an array controller according to the second aspect of the invention connected for communication to a host computer system and an array of data storage devices.

In the present invention therefore, when a write I/O request is received for a part of the array which has not been rebuilt, the write I/O is halted and the data for the area affected by the write I/O is rebuilt and stored on the replacement disk. Only once the rebuild for that area is complete does the write operation proceed.

This is in contrast to the prior art as exemplified by US 5390187 in which when a write request is received which does not involve the replacement disk, the operation that is executed in this scenario is a conventional RMW operation. As indicated in the introductory portion of the description, such an operation during a rebuild process can lead to corrupted data in the event of a power failure.

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings.

Brief Description of the Drawings

Figure 1 shows, in conceptual form, a data processing system comprising a main processing unit connected to a disk array;

Figure 2A is a diagram of an example RAID level 5 system in an initial state;

Figure 2B is a diagram of a RAID level 5 system with a failed disk member;

Figure 2C is a diagram of a RAID level 5 system with a replacement disk member prior to data reconstruction;

Figure 2D is a diagram of a RAID level 5 system with the replacement disk in partially reconstructed state;

Figure 2E is a diagram of a RAID level 5 system during a write I/O operation which does not involve the replacement disk;

Figure 2F is a diagram of a RAID level 5 system on completion of the write operation which does not involve the replacement disk; and

Figure 3 is a flow diagram showing a write I/O operation which does not involve the replacement disk.

Detailed Description of the Invention

Figure 1 is a diagram of a generalised RAID array subsystem comprising a host system 10 including a CPU 12 on which applications 14 are selectively executed. In the host system, the CPU is coupled to an array adapter 20 which provides a management function for an array 30 of disk storage devices D1, D2, D3, D4 and D5 (hereinafter referred to as disks). In Figure 1, the disks are connected for communication to the adapter in a serial loop 40 according to the Serial Storage Architecture (SSA); however the exact type of adapter to array connection employed is not critical to the invention. The adapter includes array management logic 22 for providing various services to the host system including the services necessary to manage the disks as a RAID array. Also included in the adapter for use during I/O operations between the disk array and the host system is a cache 23 for storing data, and non-volatile memory in the form of NVRAM 24 for storing metadata.

In the present embodiment, the adapter is designed to configure and manage the disk array according to level 5 of the RAID scheme i.e. as an independent access RAID array with distributed parity. It will be

appreciated however that the invention is useful at least with a RAID level 4 array i.e. independent access RAID array with parity on one disk.

The RAID level 5 array employed in the present embodiment comprises all five disks of the array. Data is stored on the array in stripes wherein a stripe comprises four blocks or strips of user data, each block being stored on a different disk (e.g. disks D1 to D4), and a parity block which is stored on a fifth disk (e.g. disk D5). The block size may be of any desired size e.g. byte, sector or multi-sector. In accordance with RAID level 5, the parity blocks of different stripes are stored on different disks.

With reference to Figures 2A to 2F and Figure 3, there will now be described the rebuild operation according to a preferred embodiment of the present invention.

Figure 2A shows an example disk array in an initial state. The disk array comprises disks D1 to D5. Each row A to F represents a stripe of data. Parity data, which is calculated as the XOR of all the user data in the stripe, is indicated by circled numbers and is distributed throughout the array. For ease of reference, one bit blocks are shown for each disk.

Figure 2B shows the same disk array as Figure 2A but with a failed disk D4. The dashed lines for the D4 blocks in stripes A to F indicate lost data.

Figure 2C shows the same disk array as Figure 2B but with a replacement disk D4' in place of failed disk D4. The crosses for the blocks in D4' indicate either random data or zero data. Once the replacement disk is substituted for the failed disk, the data reconstruction process begins. In accordance with common practice, the array management logic of the adapter is arranged to carry out the reconstruction process concurrently with other read or write I/O operations initiated by the host CPU. Conventional techniques can be used for monitoring the rebuild process. In the present embodiment, a record of the rebuild progress is made in volatile memory. For example a bitmap may be used in which each bit represents a stripe of the array. The bitmap may be checkpointed after a number of stripes have been rebuilt, so as to reduce the amount of rebuild that might have to be repeated after a power interruption.

In the present embodiment, the rebuild process begins with stripe A. The existing user data and parity blocks for stripe A are read from disks D1 to D5, and XOR'd together to generate the replacement data which is written to the appropriate location on disk D4'. The rebuild process

continues until the situation as shown in Figure 2D wherein replacement data has been written to the replacement disk for stripes A to C.

At this point it is assumed that the adapter receives a write I/O request from the host CPU which involves replacing the user data in disk 2, stripe E. It can be seen from Figure 2E that this block of data is in a stripe which has not yet been rebuilt. It can also be seen that the normal RMW operation which would be required to write this block does not involve the replacement disk.

In the prior art, the operation which is executed at this point is a conventional RMW operation involving reading the old data from disk D2 and parity from disk D1, updating the parity and rewriting the updated parity and new data to the disks. As indicated in the introductory portion of the description, such an operation during a rebuild process can lead to corrupted data in the event of a power failure.

The present invention provides for a modified operation which avoids the potential for data loss in this scenario by rebuilding the affected stripe before carrying out the requested write I/O. In a preferred embodiment, described with reference to Figure 3, the write I/O operation starts at step 100. As indicated in Figure 2E this operation involves writing a '0' to stripe E of disk D2. In step 102, this write I/O to the array is blocked to prevent execution of the standard RMW operation. The area (stripe) to be rebuilt is then identified. In step 104, the array management logic 22 of the adapter 20 reads all the existing data blocks from the affected stripe which in the present case is stripe E. In step 106, the replacement block for disk D4' is reconstructed and held in cache, along with the other stripe E data. In step 108, the reconstructed data is written to the replacement disk and at step 110, a record is made in volatile memory that the area has been rebuilt. At this point the status of the array is as indicated in Figure 2E.

Prior to recommencing the blocked write I/O, a mark is placed in NVRAM in step 112 that parity is in doubt for the stripe affected by the write I/O. In addition, a record is made, also in NVRAM, that a rebuild has been performed for the affected stripe. This record must be made non-volatile for the duration of the lifetime of the non-volatile parity in doubt mark. This can be done in one of a number of ways: (i) a separate record can be made in NVRAM which is placed before, and removed after the parity in doubt mark; (ii) a flag or separate indicator is used alongside the parity in doubt mark; or (iii) the existing parity in doubt mark is used as an implicit indicator that the rebuild has been performed.

In step 114, the new host data (in this case a '0') is fetched from the host. In step 116, a write request is issued to the disk and the new host data is written to disk D2. In step 118, the old parity (from disk D1) and the old data (from disk D2) is retrieved from the adapter cache and along with the new data (also from cache) is used to calculate the new parity in the conventional way. In step 120, the new parity is written to disk D1. Once the new parity is safely written to disk, the parity in doubt mark for the affected stripe is then removed from NVRAM. The write operation is terminated in step 124.

On completion of the write operation, the status of the array is as indicated in Figure 2 in which the updated parity and new data are shown encircled. The record made in volatile memory at step 110 that stripe E has been rebuilt (effectively out of turn) is advantageously retained during the remainder of the array rebuild process. This is so that the rebuilt area is not repeatedly rebuilt on every host write I/O, only on the first write I/O invokes the rebuild.

The normal rebuild process executes concurrently with this write operation but means are provided to lock the stripe affected by the write to prevent any attempt to otherwise rebuild the affected stripe during the write operation.

As will be appreciated from the above description, the invention provides a way of guaranteeing data integrity and reliability of data held in n array which is being rebuilt whilst subject to write I/O activity, without requiring the provision of extra hardware over that required for the RAID 5 algorithm itself.

CLAIMS

1. A method for reconstructing data from a failed data storage device to a replacement data storage device in a redundant data storage array including a plurality N of data storage devices, data being arranged on the devices in multi-block stripes each of which comprises N-1 data blocks and a parity block, with one block from each stripe being located on each of the N devices, the method comprising:

reconstructing each data block of the failed storage device for each stripe in the array and storing the reconstructed data block on the replacement storage device;

wherein in response to a write request to modify a data block which is part of a stripe which has not been reconstructed, the method comprises the further steps of:

blocking the write request;

determining the data stripe which includes the data block to be modified;

reconstructing the replacement data block for the determined data stripe;

storing the replacement data block on the replacement disk; and

subsequently executing the write request.

2. A method as claimed in claim 1 comprising the further step of, after storing the replacement data block on the replacement disk, making a record in non-volatile memory that the replacement data block for the determined data stripe has been reconstructed.

3. A method as claimed in claim 2, further comprising, after storing the replacement data block on the replacement disk, making a record in non-volatile memory that parity is in doubt for the stripe affected by the write request.

4. A method as claimed in any preceding claim wherein the step of reconstructing the replacement data block comprises reading the blocks of the determined stripe, generating the replacement data block from the read data blocks and storing the read data blocks and replacement block in a cache.

5. A method as claimed in claim 2 or claim 3 wherein the non-volatile memory comprises NVRAM.

6. An array controller for the reconstruction of data from a failed data storage device to a replacement data storage device in a redundant data storage array including a plurality N of data storage devices, data being arranged on the devices in multi-block stripes each of which comprises N-1 data blocks and a parity block, with one block from each stripe being located on each of the N devices, the controller including array management means for receiving, during the data reconstruction process, a write request to modify a data block which is part of a data stripe that has not been reconstructed, and responsive to such a request to block the write request, determine the data stripe which includes the data block to be modified; reconstruct the replacement data block for the determined data stripe; store the replacement data block on the replacement disk; and recommence the write request.

7. An array controller as claimed in claim 6 further comprising non-volatile memory, the array management means being operable to make a record in non-volatile memory that the replacement data block for the determined data stripe has been reconstructed.

8. A data storage array comprising an array controller as claimed in claim 6 or claim 7 connected for communication to a host computer system and an array of data storage devices.

9. A data storage array as claimed in claim 8 wherein the array controller is an adapter card located with the host computer system.

10. A data storage array as claimed in claim 8 or claim 9 wherein the data storage devices comprise disk storage devices.

11. A data storage array as claimed in any of claims 8 to 10 wherein the data storage devices are configured as a RAID 5 array.

The Patent Office

INVESTOR IN PEOPLE

Application No: GB 9823460.2 **Examiner:** K. Sylvan
Claims searched: 1-11 **Date of search:** 26 March 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): G4A (AME) G5R (RGB,RB33)

Int Cl (Ed.6): G06F (11/10)

Other: Online: EPODOC

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X,Y	EP0482819 A2	Array Technology. Equivalent to US5390187 acknowledged in the application. See column 5 line 53 to column 6 line 31.	X:1,6,8,10 ,11 Y: 2,5,7
Y	EP0462917 A2	IBM. See column 9 lines 29-47 and column 5 lines 51-55.	2,5,7
X	US5522031	DEC. Acknowledged in the application. See claim 1.	1,2,6,7,8, 10,11

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.